ELSEVIER

Theoretical Computer Science 209 (1998) 195–211

Theoretical Computer Science

Communication complexity of fault-tolerant information diffusion[1]

Luisa Gargano*, Adele A. Rescigno

Dipartimento di Informatica ed Applicazioni, Universita di Salerno, 84081 Baronissi (SA), Italy

Received December 1995
Communicated by G. Ausiello

Abstract

This paper considers problems of fault-tolerant information diffusion in a network with cost function. We show that the problem of determining the minimum cost necessary to perform fault-tolerant gossiping among a given set of participants is NP-hard and give approximate (with respect to the cost) fault-tolerant gossiping algorithms. We also analyze the communication time and communication complexity of fault-tolerant gossiping algorithms. Finally, we give an optimal cost fault-tolerant broadcasting algorithm and apply our results to the atomic commitment problem. © 1998 — Elsevier Science B.V. All rights reserved

1. Introduction

In this paper we study the problems of fault-tolerant broadcasting, gossiping, and atomic commitment in a weighted network.

Gossiping in an interconnection network is easily described as the process of information diffusion in which initially each participant to the process knows a block of information that has to be communicated to all the other participants by means of a sequence of message transmissions (calls). During each call the calling node sends to the receiving one every block it has collected by that time. Gossiping arises in a large class of computation problems, such as linear system solving, matrix manipulation, Discrete Fourier Transform, and sorting, where both input and output data are required to be distributed across the network [10, 36]. Due to the interesting theoretical questions it poses and its numerous practical applications, gossiping has been widely studied under various communication models [3, 6–9, 14, 18, 21, 27, 31, 34, 36, 37, 43, 46, 48]. Recent survey papers collecting the latest results are [19, 30, 32].

* Corresponding author. E-mail: lg@dia.unisa.it.

[1] Research partially supported by the Italian Ministry of University and of Scientific Research in the framework of the "Algoritmi, Modelli di Calcolo e Strutture Informative" project.

PII S0304-3975(97)00109-6

An important related problem is the *atomic commitment* problem [48]. This arises in the processing of transactions of distributed database systems. A transaction consists of several subtransactions each running at a different site having its local database. If a subtransaction is completed successfully the local manager validates the update (commit the subtransaction), otherwise the local manager invalidates it (abort the subtransaction). The generally accepted solution to the atomic commitment problem consists of: If all local managers commit the subtransaction then commit the whole transaction, otherwise abort all the updates. Therefore, the atomic commitment problem can be viewed as a gossiping problem in which the node blocks are votes (yes or no) and each individual must compute their conjunction, commit holds if and only if all votes are yes. The main difference with the gossiping problem is that whenever a node has received a no vote it already knows that the result of the conjunction is no and does not need to collect further votes. Therefore, a no voter can abort the process by disseminating an abort message; the process coincides with gossiping when all votes are yes. In the sequel we will use the terminology of gossiping, we will explicitly return to the atomic commitment problem in Section 6 to include abort instances.

To handle the case of dissemination of abort messages, we shall also consider fault-tolerant broadcasting. *Broadcasting* refers to the process in which initially one individual knows a block of information which must be communicated to every other individual by a sequence of calls. This problem has been widely studied (cf. [7, 9, 21, 22, 25, 24, 27, 38, 41–43, 46] and the surveys [19, 30, 32]). Broadcasting is a fundamental problem in the control of distributed systems and in parallel computing. For instance, in computer networks many tasks, such as scheduling, require that a processor sends a block of information to all other processors [47]. Moreover, many numerical algorithms, such as Gaussian elimination or conjugate gradient algorithm, require that some data produced at one node must be made available to all other processors in order to continue the computation (see [45] and references therein quoted).

1.1. The model

Consider a communication network modelled by a graph (V,E) where the node set V represents the set of processors of the network and E represents the set of the communication lines between processors.

Each node in V taking part in the (gossiping, broadcasting or atomic commitment) process will be called a *participant* and the set of participants will be denoted by $\mathscr{P}$. Without loss of generality, we suppose $\mathscr{P}=\{1,\dots,p\}$. We regard the set $\mathscr{P}$ as the node set of a complete digraph $K_{\mathscr{P}}=(\mathscr{P},A_{\mathscr{P}})$, with $A_{\mathscr{P}}=\{(i,j)\colon i,j\in\mathscr{P},\ i\neq j\}$. Each arc $(i,j)\in A_{\mathscr{P}}$ is labelled with the cost $c(i,j)>0$ of sending a message along the interconnection network from the participant i to the participant j, with $c(i,j)=c(j,i)$. The communication costs may differ from one arc to another; for example, the communication cost $c(i,j)$ may represent the distance between the node i and the node j in the interconnection network. Motivations to study the communication load by means of this weighted graph model vs the traditional study of the number of intersite messages are given by Wolfson and Segall in [48], see also [23, 35]. The *communication cost* of an algorithm is the total cost of the arcs the algorithm uses to send messages among participants.

The *communication time* of an algorithm is the interval of time necessary for the completion of the algorithm itself. To each arc $(i,j) \in A_{\mathscr{P}}$ it is associated the *travel time* $t(i,j)$ needed for a message from the participant i to reach the participant j. Notice that since our aim will be to overcome the effect of transmission failures, a minimum of synchronism appears necessary in order to perform gossiping; the travel time $t(i,j)$ represents the maximum amount of time needed for a non faulty transmission of an unitary file from i to j. However, our broadcasting algorithms will work also in a completely asynchronous system.

The *communication complexity* of an algorithm is defined as the product of its communication cost and its communication time.

1.2. Our results and related works

As the number of components of a distributed system increases, the probability that some component works incorrectly becomes non-negligible. Therefore, the incorporation of some redundancy in the basic primitives performed in the network (such as gossiping, commit and broadcasting) is necessary. These problems have been widely studied under various models, see the recent survey by Pelc [40].

In this paper we investigate the minimum necessary communication complexity of algorithms tolerating transmission failures, i.e. algorithms that complete their task even though some messages may fail to reach (on time) their destinations. A gossiping/commitment/broadcasting protocol is called k-fault tolerant (k-ft) if any strategy for suppressing up to k calls cannot prevent a successful termination of the protocol; It is important to point out that we consider algorithms in which the sequence of calls is fixed and cannot be changed if faults are detected (cf. [1, 7, 21, 22, 24, 25, 27, 38, 42] and the survey [40]).

Communication protocols in the model considered in this paper have been studied in [23, 44, 48]. Minimum-cost gossiping under the assumption of the “telephone model” of communication (cf. [30]) was considered in [13, 35]. Notice that the well studied problem of gossiping with the minimum number of calls (cf. [4, 7, 11, 28, 5, 33, 29] and the survey [30]) is a particular case of gossiping minimizing the cost when all calls have the same cost.

Recently, Wolfson and Segall gave a solution to the interesting problem of performing gossiping (commitment) in weighted networks [48]. They show that the minimum cost of an algorithm that solves the gossiping or commit problem is equal to 2·(*cost of a minimum spanning tree of* $K_{\mathscr{P}}$). Moreover, they propose an algorithm that achieves this bound and show that its communication time (resp. complexity) cannot be worse than $|\mathscr{P}|$ times the minimum communication time (complexity). The results by Wolfson and Segall are restricted to the case in which no faults occur while the protocol is executed. They state as open the important problem of handling failures.

In this paper we study fault-tolerant information diffusion problems in the model introduced in [48] and studied, among the others, in [23, 35]. In Section 3 we study k-ft broadcasting algorithms. We lower bound the minimum communication cost of k-ft broadcasting by $(k+1) \cdot$ (*cost of a minimum spanning tree of* $K_{\mathscr{P}}$) and give an optimal k-ft broadcasting algorithm. In Section 4 we study the communication cost of fault tolerant gossiping algorithms. In particular, we show that the problem of determining the minimum cost necessary to perform k-ft gossiping among a given set of participants is NP-hard, for each $k \geqslant 1$. This should put in contrast with the result in [48] corresponding to the case $k=0$, i.e., when no faults are assumed, for which a polynomial time algorithm exists. In Section 5 we give approximate (with respect to the cost) fault-tolerant gossiping algorithms. The communication cost of our algorithm is upper bounded by twice the cost of an optimal one. Moreover, in case of uniform cost function, our algorithm uses the minimum number of calls necessary to perform k-ft gossiping among p participants, that is, is $(k+2)p-2$ [7]. We also show that the communication time of our k-ft gossiping algorithm is at most a factor $k(p-1)+2p-1$ larger than that of an optimal algorithm; moreover, its communication complexity cannot be worse than $2k(p-1)+4p-2$ times the minimum communication complexity. Notice that, generally, the same algorithm cannot minimize both cost and time and that minimizing the communication complexity is a NP-complete problem even in the absence of faults [48]. In Section 6 we apply our results to the atomic commitment problem.

2. Multi-digraph associated to a communication protocol

We introduce here the notion of multi-digraph associated to an instance of a (gossiping, broadcasting, or commitment) protocol that will be used in the following sections.

The sequence of calls of an instance I of a protocol P will be represented by a labelled multi-digraph $P(I)=(\mathscr{P},A)$ having as node set the set $\mathscr{P}$ of participants, as arc set the multiset A in which each arc (i,j) represents a message sent from i to j, and arc labels represent the temporal order in which calls are made. We call two paths (*transmission*) *disjoint* if they share no arc-label pair; two disjoint paths α_1 and α_2 can both contain the arc (i,j) only if the label of (i,j) in α_1 is different from the label of (i,j) in α_2, i.e. the calls from i to j are made at different times. A path from i to j is called *ascending* if the sequence of labels is strictly increasing when moving from i to j. Since a participant j receives the block of participant i only if $P(I)$ contains an ascending path from i to j and since one cannot know a priori which calls will fail, an instance can tolerate up to k failures if and only if for each choice of k arc-label pairs of the multi-digraph $P(I)$ the multi-digraph $P'(I)$ obtained from $P(I)$ by removing these arc-label pairs contains at least one ascending path from i to j, for each $i,j \in \mathscr{P}$ for which the protocol requires that the block originated at i reaches j. The cost of the instance I (or equivalently, the cost of the associated multi-digraph $G(I)$) is then the sum of all arcs of $G(I)$, each added as many times as its multiplicity.

```
BROADCASTING(c, P, k) \ at a participant i ∈ P
1. Construct a minimum spanning tree T of the cost graph K_P.
2. If source then send the message k + 1 times to each neighbor in T.
        else  as soon as a copy of the message is receive d from a neighbor
              forward this copy (separately from others) to all the other neighbors.
```

Fig. 1.

3. Broadcasting

Broadcasting refers to the process of information diffusion in which one individual, called the *source*, knows one block (of information) that must be communicated to each of the other participant in the set $\mathscr{P}$. If a broadcasting instance has to tolerate up to k transmission failures then at least $k+1$ disjoint ascending paths from the source to each other participant must be created. An algorithm can be obtained by considering a minimum spanning tree T of $K_{\mathscr{P}}$ and using $k+1$ times the edges of T, directed away from the source. This is done by the algorithm in Fig. 1.

Let $cost(ST_{\min})$ represent the cost of the minimum spanning tree of the cost graph $K_{\mathscr{P}}$.

Theorem 3.1. *Let $\mathscr{P}$ be a set of participants, and let c be a set of associated communication costs. The minimum communication cost of a k-ft broadcasting instance is $(k+1)\cdot cost(ST_{\min})$.*

Proof. The algorithm BROADCASTING$(c,\mathscr{P},k)$ gives an upper bound of $(k+1)\cdot cost(ST_{\min})$ on the communication cost of a k-ft broadcasting instance. We prove now that this cost is minimum. Denote by x the source of the broadcast. Consider an instance I and denote by $B(I)=(\mathscr{P},A(B(I)))$ its associated multi-digraph. If we suppose that $B(I)$ does not contain $k+1$ disjoint spanning trees oriented away from the source then by Edmonds theorem [16] we get that there exists $X\subset\mathscr{P}$ with $x\in X$ such that at most k arcs are directed from a node in X to a node in $\mathscr{P}-X$. Therefore, for any $i\in\mathscr{P}-X$ the multi-digraph $B(I)$ can contain at most k disjoint paths from x to i, contradicting the hypothesis that I is a k-ft broadcasting instance. □

The algorithm MULTI-BROADCASTING$(c,\mathscr{P},k)$ in Fig. 2 will be needed in Section 6. It generalizes BROADCASTING$(c,\mathscr{P},k)$ to the case when initially more sources know a same message that has to be broadcasted (participants do not know the identity of the sources). An upper bound on the cost of the algorithm MULTI-BROADCASTING$(c,\mathscr{P},k)$ is $(k+1)cost(ST_{\min})$.

4. Gossiping

In the gossiping process each participant $i\in\mathscr{P}$ has a block (of information) that needs to be communicated to all the other participants. During each call the calling

MULTI-BROADCASTING$(c, \mathcal{P}, k)$ \ at a participant $i \in \mathcal{P}$
1. Construct a minimum spanning tree T of the cost graph $K_{\mathcal{P}}$.
2. If source then send the message $k+1$ times to each neighbor in T.
   else Let j be the neighbor from which the first copy of the message is received
   each time a copy of the message is received from j forward this copy (separately)
   to all the other neighbors but those from which some copy of the message is arrived.

Fig. 2.

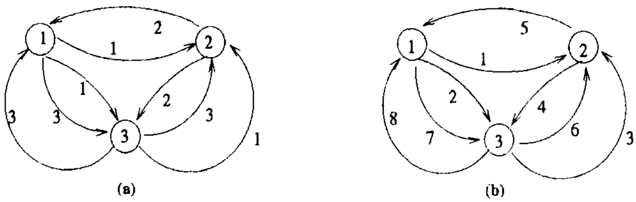

Fig. 3.

processor sends to the receiving one a message containing every block it has collected by the time of the call.

As customary, we assume that blocks can be combined so that any call from a participant i to a participant j can be considered of the same cost $c(i,j)$. The time needed for combining is irrelevant and treated as zero. This is indeed the case in many applications as the commitment problem considered in this paper. We recall that we consider algorithms in which the sequence of calls is fixed and cannot be changed if faults are detected.

Denote by $G(I)$ the multi-digraph associated to an instance I of a gossiping algorithm. A labelled multi-digraph G is called k-ft gossiping multi-digraph if $G = G(I)$ for some k-ft gossiping instance I.

Example 1. Let us consider the set $\mathscr{P} = \{1,2,3\}$. Two possible 1-ft gossiping multi-digraphs are given in Fig. 3. Each contains two disjoint ascending paths from i to j for each $i,j \in \mathscr{P}$, $i \neq j$.

By Menger's theorem (see [39, Problem 6.39])

Lemma 4.1. *A gossiping algorithm instance can tolerate up to k failures if and only if for each $i,j \in \mathscr{P}$ with $i \neq j$, the multi-digraph $G(I)$ contains $k+1$ disjoint ascending paths from i to j.*

4.1. Minimum communication cost of fault-tolerant gossiping

In this section we study the *minimum communication cost* of fault tolerant gossiping algorithms. We prove that finding the minimum cost of a fault tolerant gossiping multi-digraph is NP-hard. To this aim we need some results on the minimum number of calls of k-ft gossiping algorithms. Berman and Hawrycz proved that

Lemma 4.2 (Berman and Hawrycz [7]). *The minimum number of calls of any k-ft gossiping algorithm among p participants is* $(k+2)p-2$.

The following lemma is crucial.

Main lemma. *Let I be a k-ft gossiping algorithm instance on the participant set $\mathscr{P}$ that uses $(k+2)|\mathscr{P}|-2$ calls and let $G(I)=(\mathscr{P},A(G(I)))$ be its associated multi-digraph. If $k\geqslant 1$ and $|\mathscr{P}|\geqslant 3$ then the undirected graph $G_U(I)=(\mathscr{P},E(G_U(I)))$ with*

$$E(G_U(I))=\{\{i,j\}:\ (i,j)\in A(G(I))\ \text{or}\ (j,i)\in A(G(I))\}$$

contains an Hamiltonian circuit.

Before giving the proof of the Main lemma we derive its consequences: The problem of finding the minimum communication cost of a fault tolerant gossiping multi-digraph is NP-hard. Indeed, consider the following associated decision problem:

k-GOSSIPING-MULTI-DIGRAPH
INSTANCE: A cost graph $K_{\mathscr{P}}$ and a bound $B>0$;
QUESTION: Is there a k-ft gossiping multi-digraph of cost $\leqslant B$?

Main corollary. *k-GOSSIPING-MULTI-DIGRAPH is NP-hard for any $k\geqslant 1$.*

Proof. We can reduce an instance of the Hamiltonian circuit problem, which is NP-complete [20], on a graph $H=(\mathscr{P},E(H))$ into an instance of the above k-GOSSIPING-MULTI-DIGRAPH decision problem in which the cost graph $K_{\mathscr{P}}$ has costs

$$c(i,j)=\begin{cases}1 & \text{if } \{i,j\}\in E(H)\\ 2 & \text{if } \{i,j\}\notin E(H)\end{cases}$$

and the bound is $B=(k+2)|\mathscr{P}|-2$. If there exists a k-ft gossiping multi-digraph $G(I)$ on the set of participants $\mathscr{P}$ of cost $\leqslant(k+2)|\mathscr{P}|-2$ then, by Lemma 4.2, it must contain exactly $(k+2)|\mathscr{P}|-2$ arcs of cost 1, i.e., $A(G(I))$ consists only of arcs (i,j) with $\{i,j\}\in E(H)$. Therefore, $E(G_U(I))\subseteq E(H)$ and, by the Main lemma, also H contains an Hamiltonian circuit.

On the other hand, if H contains an Hamiltonian circuit, say $\{i_1,i_2\},\{i_2,i_3\},\ldots,\{i_{p-1},i_p\},\{i_p,i_1\}$, following [7], we can easily obtain a k-ft gossiping multi-digraph of cost $(k+2)p-2$ as follows:

For $t=1,\ldots,p-2$: join i_t to i_{t+1} with $k+2$ arcs directed toward i_{t+1}; give to these $k+2$ arcs temporal labels, respectively, $t,p+t,2p+t,\ldots,(k+1)p+t$.
Join i_{p-1} to i_p with $k+1$ arcs directed toward i_p; give to these $k+1$ arcs temporal labels, respectively, $p-1,2p-1,\ldots,(k+1)p-1$.

Join i_p to i_1 with $k+1$ arcs directed toward i_1; give to these $k+1$ arcs temporal labels, respectively, $p, 2p, \ldots, (k+1)p$. □

4.2. Proof of main lemma

In order to prove the Main lemma, let us first notice that since we are not concerned here with the time that is necessary to complete the gossiping instance, we can assume that no two calls are made at the same time. This implies that all the calls of the instance I (i.e., arcs of the associated multi-digraph $G(I)$) have different temporal labels. It is easy to see that we can always modify the label $\ell(a)$ of each arc a in $G(I)$, in order to obtain new labels, say $\ell'(a)$, so that $\ell'(a) \neq \ell'(a')$ for each $a \neq a'$ and, in order to preserve the ascending paths, $\ell'(a) < \ell'(a')$ whenever $\ell(a) < \ell(a')$. We stress that such an assumption does not influence the object of our analysis, that is, the arc multiset of $G(I)$.

Example 4.1 (*continued*). In order to fullfill the above assumption we can modify the temporal labels of the graph in Fig. 3(a) as shown in Fig. 3(b). It is immediate to see that ascending paths are not influenced by such a time expansion.

We first need to introduce some notation.

We denote by $p = |\mathscr{P}|$ the number of participants and by $T = (k+2)p - 2$ the number of calls made by the algorithm.

For $t = 1, \ldots, T$, we denote by s_t the sender of the tth call and by r_t the recipient of the tth call. Therefore, the multi-digraph $G(I)$ associated to the instance I is the multi-digraph on node set $\mathscr{P}$ with arcs (s_t, r_t) with label t, for $t = 1, \ldots, T$.

We denote by G_t, for $1 \leqslant t \leqslant T$, the multi-digraph on node set $\mathscr{P}$ with arcs $(s_1, r_1), \ldots, (s_t, r_t)$, and by G_{t_1,t_2}, for $1 \leqslant t_1 < t_2 \leqslant T$, the multi-digraph on node set $\mathscr{P}$ with arcs $(s_{t_1}, r_{t_1}), \ldots, (s_{t_2}, r_{t_2})$.

Given $i \in \mathscr{P}$, let $d_t^{\text{out}}(i)$ denote the outdegree of i in G_t.

Finally, given $i, j \in \mathscr{P}$ and $1 \leqslant t \leqslant T$ let $N_t(j,i)$ denote the number of disjoint ascending paths from j to i in G_t and let

$$N_t(i) = \min_{j \in \mathscr{P}, j \neq i} N_t(j,i) \quad \text{and} \quad N_t = \sum_{i \in \mathscr{P}} N_t(i). \tag{1}$$

Notice that the k-ft multi-digraph $G(I) = G_T$ must contain at least $k+1$ disjoint ascending paths from j to i, for each $j, i \in \mathscr{P}$, that is,

$$N_T(i) \geqslant k+1 \tag{2}$$

for each $i \in \mathscr{P}$. Moreover, for each t the number of disjoint ascending paths from j to i in G_t is upper bounded by the outdegree of j in G_t, thus

$$d_t^{\text{out}}(j) \geqslant N_t(i) \tag{3}$$

for each $t = 1, \ldots, T$ and $i, j \in \mathscr{P}$ with $i \neq j$.

We will prove that the undirected graph corresponding to $G_{p,2p-1}$ is a cycle on $\mathscr{P}$. To this aim we need the following technical intermediate results.

Fact 4.1.

$$N_t = \begin{cases} 0 & \text{if } 1 \leqslant t \leqslant p-2, \\ t-p+2 & \text{if } p-1 \leqslant t \leqslant T; \end{cases} \tag{4}$$

$$N_t(i) = \begin{cases} N_{t-1}(i) & \text{if } i \neq r_t, \\ N_{t-1}(i)+1 & \text{if } i = r_t; \end{cases} \tag{5}$$

for each $i \in \mathscr{P}$ *and* $p-1 \leqslant t \leqslant T$.

Proof. Since G_t contains p nodes and t arcs we get that for each $t \leqslant p-2$ there are at least two nodes $j_1, j_2 \in \mathscr{P}$ whose outdegree in G_t is 0. Therefore, by (3) we have that for each $i \in \mathscr{P}$

$$N_t(i) \leqslant \min_{j \in \mathscr{P}, j \neq i} d_t^{\text{out}}(j) = 0,$$

that implies $N_t = \sum_{i \in \mathscr{P}} N_t(i) = 0$, for $t = 1, \ldots, p-2$. Consider now $t \geqslant p-1$. The graph G_t differs from G_{t-1} only in the arc (s_t, r_t); therefore, this arc cannot appear in G_t on any ascending path going to any $i \neq r_t$ and we get

$$N_{t-1}(r_t) \leqslant N_t(r_t) \leqslant N_{t-1}(r_t)+1 \quad \text{and} \quad N_t(i) = N_{t-1}(i), \text{ for each } i \neq r_t. \tag{6}$$

Therefore,

$$N_t = \sum_{i \in \mathscr{P}} N_t(i) \leqslant \sum_{i \in \mathscr{P}} N_{t-1}(i) + 1 = N_{t-1} + 1. \tag{7}$$

By iterating (7) we have

$$N_t \leqslant N_{p-2} + t - (p-2) = t - p + 2.$$

In order to get (4) we observe that by iterating (7) we have $N_T \leqslant N_t + T - t$ that, by (2), gives

$$N_t \geqslant N_T - T + t = \sum_{i \in \mathscr{P}} N_T(i) - T + t \geqslant p(k+1) - ((k+2)p-2) + t = t - p + 2.$$

Equality (5) follows by noticing that (6) and (4) imply $N_t(r_t) - N_{t-1}(r_t) = N_t - N_{t-1} = 1$. $\square$

Fact 4.2. *In* G_{p-1} *each* $j \in \mathscr{P}$ *has outdegree*

$$d_{p-1}^{\text{out}}(j) = \begin{cases} 1 & \text{if } j \neq r_{p-1}, \\ 0 & \text{if } j = r_{p-1}. \end{cases} \tag{8}$$

Proof. By the definition of N_{p-2} and (4) we have $N_{p-2}(j)=N_{p-2}=0$, for each $j\in\mathscr{P}$. Moreover by (5) we have $N_{p-1}(j)=N_{p-2}(j)=0$ if $j\neq r_{p-1}$ and $N_{p-1}(r_{p-1})=N_{p-2}(r_{p-1})+1=1$. Therefore, by (3) we get that for each $j\neq r_{p-1}$

$$d_{p-1}^{\mathrm{out}}(j)\geqslant \max_{i\in\mathscr{P},i\neq j} N_{p-1}(i)=1.$$

Since G_{p-1} has exactly $p-1$ arcs, we have (8). □

Fact 4.3. *For each* $i\in\mathscr{P}$

$$N_{2p-2}(i)=1, \tag{9}$$

$$N_{2p-1}(i)=\begin{cases}1 & \textit{if } i\neq r_{2p-1},\\ 2 & \textit{if } i=r_{2p-1}.\end{cases} \tag{10}$$

Proof. We first show that each node has outdegree at least 1 in G_{2p-2}. By (8) we know that $d_{2p-2}^{\mathrm{out}}(i)\geqslant d_{p-1}^{\mathrm{out}}(i)=1$ for each $i\neq r_{p-1}$. If we suppose that $d_{2p-2}^{\mathrm{out}}(r_{p-1})=0$ then we have $N_{2p-2}(i)=0$ for each $i\neq r_{p-1}$; therefore

$$p=N_{2p-2}=N_{2p-2}(r_{p-1})\leqslant d_{2p-2}^{\mathrm{out}}(i), \tag{11}$$

for each $i\neq r_{p-1}$. Since G_{2p-2} has exactly $2p-2$ arcs, inequality (11) implies that $2p-2=\sum_{i\in\mathscr{P}} d_{2p-2}^{\mathrm{out}}(i)\geqslant p(p-1)$, which is impossible for any $p\geqslant 3$. Therefore, in G_{2p-2} each node has outdegree at least 1; in order to have $2p-2$ arcs there must exist at least two nodes with outdegree equal to 1. By (3) we get

$$N_{2p-2}(i)\leqslant \min_{j\in\mathscr{P},j\neq i} d_{2p-2}^{\mathrm{out}}(j)=1,$$

for each $i\in\mathscr{P}$. The equality (9) follows by noticing that by (4) we have $\sum_{i\in\mathscr{P}} N_{2p-2}(i)=N_{2p-2}=p$. In order to prove (10) we observe that (5) implies $N_{2p-1}(i)=N_{2p-2}(i)=1$, for each $i\neq r_{2p-1}$ and $N_{2p-1}(r_{2p-1})=N_{2p-2}(r_{2p-1})+1=2$. □

Fact 4.4. $G_{p,2p-1}$ *contains*:

(a) *an ascending path from* r_{p-1} *to each other node* $i\in\mathscr{P}$ *with* $i\neq r_{p-1}$;

(b) *an ascending path from* i *to* r_{2p-1}, *for each* $i\in\mathscr{P}$ *with* $i\neq r_{2p-1}$.

(c) *two disjoint ascending paths from* r_{p-1} *to* r_{2p-1}, *if* $r_{p-1}\neq r_{2p-1}$.

Proof. By (10) we have $N_{2p-1}(i)\geqslant 1$, for each $i\in\mathscr{P}$. This implies that G_{2p-1} contains an ascending path from r_{p-1} to each $i\neq r_{p-1}$. Since (8) tells us that $d_{p-1}^{\mathrm{out}}(r_{p-1})=0$, we can conclude that the above paths lie entirely in $G_{p,2p-1}$. Hence (a) holds.

We show now (b) and (c). Fix any $i\neq r_{2p-1}$. By (10) we have $N_{2p-1}(r_{2p-1})=2$. This implies that G_{2p-1} contains two disjoint ascending paths from i to r_{2p-1}.

If $i\neq r_{p-1}$, we know by (8) that $d_{p-1}^{\mathrm{out}}(i)=1$ and we can conclude that one of the ascending paths from i to r_{2p-1} lies entirely in $G_{p,2p-1}$; If $i=r_{p-1}$ with $r_{p-1}\neq r_{2p-1}$ we know by (8) that $d_{p-1}^{\mathrm{out}}(i)=0$ and we can conclude that both ascending paths from r_{p-1} to r_{2p-1} lie entirely in $G_{p,2p-1}$. □

In order to conclude the proof of the Main lemma, let us denote by H the undirected graph underlying $G_{p,2p-1}$, that is, $H=(\mathscr{P},E(H))$ with

$$E(H)=\{\{s_t,r_t\}\mid t=p,\ldots,2p-1\}.$$

From Fact 4.4, we get that H is connected and each node in H has at least two incident edges. Since H has p nodes and p edges, we get that H is a connected graph with each node of degree exactly 2. Therefore, H is an Hamiltoniam circuit on $\mathscr{P}$. Since $E(H)\subseteq E(G_U(I))=\{\{s_t,r_t\}\mid t=1,\ldots,T\}$ the Main lemma follows.

5. Gossiping algorithms

We showed in Section 2.1 that computing the minimum communication cost of a fault tolerant gossiping instance is an NP-hard problem. This suggests that if a fast algorithm is desired then we must relax the request for optimality and look for approximate fault tolerant gossiping algorithms.

5.1. Biconnected spanning multi-digraphs

Definition 5.1. A biconnected spanning multi-digraph (BSM) of $\mathscr{P}$ is a multi-digraph $B=(\mathscr{P},A(B))$ having two, not necessarily distinct, nodes $\sigma,\sigma'\in\mathscr{P}$ such that
- for each $i\in\mathscr{P}$ and $\sigma\neq i\neq\sigma'$, there exist a path from σ to i and a path from i to σ',
- if $\sigma\neq\sigma'$ then there exist at least two disjoint paths from σ to σ'.

The nodes σ and σ' are called, respectively, the source and the sink of B.

Let B_0 be a spanning tree of $K_{\mathscr{P}}$ rooted in a node σ_1 and having the arcs oriented towards the root and let B_{k+1} be a spanning tree of $K_{\mathscr{P}}$ rooted in a node σ_{k+1} and having the arcs oriented towards the leaves. Moreover, let B_i, for $i=1,\ldots,k$, be k not necessarily different BSM of $K_{\mathscr{P}}$ such that B_i has source σ_i and sink σ_{i+1}, that is, the sink of B_i is the source of B_{i+1}. The desired algorithm consists of $k+2$ temporally ordered rounds: in round 0, which uses the arcs of the tree B_0, blocks from all nodes in $\mathscr{P}$ accumulate in σ_1; round r, for $r=1,\ldots,k$, uses the arcs of the BSM B_r so that both the source σ_r broadcasts all blocks it knows to each other participant and all blocks accumulate in the sink σ_{r+1}; in the last round, which uses the arcs of the tree B_{k+1}, the root σ_{k+1} broadcasts all blocks it knows to each other participant. In terms of associated multi-digraph $G(B_0,B_1,\ldots,B_{k+1})$, this contains the arcs of each B_i, $0\leqslant i\leqslant k+1$; arcs are assigned temporal labels such that each directed path is ascending, this implies that $label(a) < label(a')$, for each $a\in A(B_i)$, $a'\in A(B_{i+1})$.

Consider any set of k arc-label pairs of $G(B_0,B_1,\ldots,B_{k+1})$ and denote by $G'(B_0,B_1,\ldots,B_{k+1})$ the multi-digraph obtained from $G(B_0,B_1,\ldots,B_{k+1})$ by removing these arc-label pairs. There exist at least two rounds which are not affected by such a removal,

APPROX-BSM($c, \mathcal{P}$)
1. Construct a minimum spanning tree T of the cost graph $K_{\mathcal{P}}$ and let $i_1 \in \mathcal{P}$ be its root.
2. Let $L = (i_1, i_2, \dots, i_p)$ be the list of nodes visited in a preorder tree walk on T and $i_{p+1} = i_1$.
3. Let $B = (\mathcal{P}, A(B))$ be the BSM obtained in the following way:
   for $t = 1, \dots, p$
   connect i_t and i_{t+1} making the minimum cost choice between the arc (i_t, i_{t+1}) and the path connecting i_t and i_{t+1} in T, directed from i_t to i_{t+1};

Fig. 4.

that is, there exist r and t, with $0 \leqslant r < t \leqslant k+1$, such that all the arcs of B_r and B_t are present in $G'(B_0, B_1, \dots, B_{k+1})$. From this it is easy to derive that for each $i, j \in \mathcal{P}$ the multigraph $G'(B_0, B_1, \dots, B_{k+1})$ contains at least one ascending path from i to j. Therefore, we have the following:

Lemma 5.1. *$G(B_0, B_1, \dots, B_{k+1})$ is a k-ft gossiping multi-digraph, for each $k \geqslant 0$.*

Since we want to minimize the cost, we build B_0 and B_{k+1} starting from a minimum spanning tree of the cost graph $K_{\mathcal{P}}$ and we choose each B_i as a BSM of $K_{\mathcal{P}}$ having as small cost as possible. Since the problem of determining a minimum BSM is NP-complete (a reduction from the Hamiltonian circuit can be easily proved following [17]), we concentrate on approximation algorithms.

We give now an algorithm to construct an approximate BSM (Fig. 4) of the cost graph $K_{\mathcal{P}}$, for a given set of participants $\mathcal{P}$ and cost function $c(i,j) = c(j,i)$, $i \neq j$.

Note that, in APPROX-BSM($c, \mathcal{P}$), if the triangle inequality holds for the cost function we can always choose the arc (i_t, i_{t+1}) and get $A(B) = \{(i_t, i_{t+1}) : 1 \leqslant t \leqslant p-1\} \cup \{(i_p, i_1)\}$.

It is easy to see that APPROX-BSM($c, \mathcal{P}$) returns a BSM whose cost is at most twice the cost of a minimum spanning tree; this can be proved along the same lines of Theorem 37.2 of [12].

5.2. Approximate gossiping algorithm

We assume, as in other papers (see [48]), that each participant knows the identity of all the other participants and the associated set of communication costs. The BSM-GOSSIPING($c, \mathcal{P}, k$) given in Fig. 5 is executed by each participant; the spanning tree T and the BSM B are identical at all the participants, this will be the case if the construction procedures are identical at all participants. *LABEL*($B_0, B_1, \dots, B_{k+1}$) represents an (easy to derive) algorithm that labels the arcs of $G(B_0, B_1, \dots, B_{k+1})$ so that *label*(i,j) is the exact time at which i sends its packet to j, considering a fail-safe message taking at most time $t(i,j)$ to go from i to j and that each participant knows the upper bound τ_i on the time at which i learns its own block, for each $i \in \mathcal{P}$.

```
BSM-GOSSIPING(c,P,k) \at participant i
1. Construct a minimum spanning tree T of the cost graph K_P.
2. Construct an approximate BSM B of the cost graph K_P and call σ its source and σ' its sink
3. B_0 is obtained by rooting T in the source σ of B_1 and directing its edges toward σ and
   Each B_i, 1 ≤ i ≤ k is obtained from B by taking source σ (resp. σ') and sink σ' (σ) if i is odd (even)
   and directing its arcs from the source to the sink.
   B_{k+1} is obtained by rooting T in the sink of B_k and directing its edges away from the root.
4. Apply the labelling LABEL(B_0, B_1, ..., B_{k+1}).
5. For t = 0 to k + 1 do
      as all the packets from incoming calls (arcs) in B_t (in B_{t-1} if source of B_t) are arrived
      or the maximum waiting time (i.e., the label of the outgoing arcs in B_t) is elapsed
      send along the outgoing arcs in B_t a packet containing all the blocks already known.
```

Fig. 5.

5.3. Communication cost of the algorithm BSM-GOSSIPING$(c,\mathscr{P},k)$

Let $cost(BSM_{\text{app}})$ and $cost(ST_{\min})$ represent, respectively, the cost of the BSM given by the approximation algorithm used and of the minimum spanning tree of the cost graph $K_{\mathscr{P}}$.

Theorem 5.1. *The cost required by the BSM gossiping algorithm is* $2 \cdot cost(ST_{\min}) + k \cdot cost(BSM_{\text{app}})$.

Corollary 5.1. *The cost of any instance of the BSM gossiping algorithm satisfies*

$$cost(I) = 2 \cdot cost(ST_{\min}) + k \cdot cost(BSM_{\text{app}}) \leqslant 2\, cost(I_{c\min}), \tag{12}$$

where $I_{c\min}$ *represents an instance of minimum possible cost.*

Proof. If we fix any participant $i \in \mathscr{P}$, the gossiping process must create at least $k+1$ disjoint paths from i to each other participant. It was shown in Theorem 3.1 that the minimum cost necessary to create such disjoint paths is $(k+1) \cdot cost(ST_{\min})$. Hence, the minimum cost of an instance is $cost(I_{c\min}) \geqslant (k+1) \cdot cost(ST_{\min})$. On the other hand by using the algorithm APPROX-BSM$(c,\mathscr{P})$ we have $cost(BSM_{\text{app}}) \leqslant 2 \cdot cost(ST_{\min})$ and (12) holds. □

Remark 5.1. We remark that by repeating $k+1$ times an optimal non-ft gossiping algorithm [48], one obtains a k-ft gossiping algorithm of cost exactly $2\,(k+1) \cdot cost(ST_{\min})$. However, in practice the algorithm based on BSMs can have a much better cost and in many cases the BSM may have "almost" the same cost as $ST_{\min}$. In particular, in case of uniform cost function with $c(i,j) = c$, the BSMs given by APPROX-BSM(c,P) are cycles and the total number of arcs used is $(k+2)p-2$; notice that by Lemma 4.2, this is the minimum possible. Therefore, the cost of BSM-GOSSIPING$(c,\mathscr{P},k)$ is the minimum possible cost $((k+2)p-2)c$, while repeating $k+1$ times a non-ft gossiping algorithm would give an algorithm of cost $2(k+1) \cdot cost(ST_{\min}) = 2(k+1)(p-1)c$. Even though we did not find a formal evidence, we are convinced that BSMs play a basic role in the fault tolerant gossiping problem and that the algorithm

BSM-GOSSIPING($c, \mathscr{P}, k$) would be optimal whenever the BSMs are. Indeed, we are tempted to put forward the following.

Conjecture The cost of a k-ft gossiping instance is lower bounded by $2\,cost(ST_{\min}) + k\,cost(BSM_{\min})$, where $cost(BSM_{\min})$ represents the minimum cost of a BSM of the cost graph $K_{\mathscr{P}}$.

5.4. Communication complexity of fault-tolerant gossiping

The communication time of an instance I on a set $\mathscr{P}$ of participants is defined as the minimum time required to perform the calls in the order specified by labelling of the associated multi-digraph $G(I)$ with respect to the set $\tau = \{\tau_i\colon i \in \mathscr{P}\}$, where τ_i is the time needed for i to have its block ready, and the set of travel times $t = \{t(i,j)\colon i,j \in \mathscr{P},\ i \neq j\}$. As in [48] we assume that t satisfies the triangle inequality. The following lower bound comes from [48]

$$time(I) \geqslant \max_{i,j\in\mathscr{P}}\{\tau_i + t(i,j)\}. \tag{13}$$

We consider now the communication time of an instance I of the algorithm BSM-GOSSIPING($c, \mathscr{P}, k$).

Theorem 5.2. *Let $\mathscr{P}$ be a set of participants. Denote by $I_{t\,\min}$ a k-ft gossiping instance which has minimum communication time and by I the instance executed by BSM-GOSSIPING($c, \mathscr{P}, k$). Then $time(I)/time(I_{t\,\min}) \leqslant k(p-1) + 2p - 1$.*

Proof. Define the time of a path α as the sum of the times $t(i,j)$ over all arcs (i,j) of α. Notice that there are at most $p-1$ arcs on a path both in the tree B_0 and the tree B_{k+1}. Moreover, we can always choose the sources and the sinks of a given BSM so that there are at most $p-1$ arcs on a path from the source to the sink in B_l, $l = 1, \ldots, k$. So the time of a path from a vertex i to any other vertex is at most $time(I) \leqslant \tau_i + (k+2)(p-1)\max_{(q,r)} t(q,r)$, where the maximum is taken over all arcs (q,r) in $B_0, B_1, \ldots, B_{k+1}$. From this and (13)

$$\frac{time(I)}{time(I_{t\,\min})} \leqslant \frac{(k+2)(p-1)t(q,r) + \tau_i}{\max_{s,j\in\mathscr{P}}\{\tau_s + t(s,j)\}} \leqslant (k+2)(p-1) + 1,$$

and the latter inequality is true since $\max_{s,j\in\mathscr{P}}\{\tau_s + t(s,j)\} \geqslant \max\{\tau_i, t(q,r)\}$. $\square$

The communication complexity of an instance I is $comm(I) = cost(I) \cdot time(I)$. Note that minimizing the communication complexity is NP-complete even in the absence of faults [48].

Theorem 5.3. *Denote by $I_{\min}$ a k-ft gossiping instance which has minimum commun-cation complexity and by I the instance executed by BSM-GOSSIPING($c, \mathscr{P}, k$).*

Then

$$comm(I)/comm(I_{\min}) \leqslant 2(k(p-1)+2p-1).$$

Proof. By Theorem 5.2 and (12) we have

$$\frac{comm(I)}{comm(I_{\min})} = \frac{time(I)\cdot cost(I)}{time(I_{\min})\cdot cost(I_{\min})} \leqslant (k(p-1)+2p-1)\frac{cost(I)}{cost(I_{c\,\min})}$$
$$\leqslant 2(k(p-1)+2p-1). \qquad \square$$

6. Atomic commitment

In this section we specifically consider the problem of atomic commitment. In this case each participant has to communicate to all the others its local vote about the completion of a transaction (yes or no), and each individual must compute the conjunction of all decisions, commit holds if all votes are yes, otherwise each participant must end the process with an abort decision. Therefore, whenever a node has received an abort vote it already knows the result of the conjunction be abort and does not need to collect further votes. A no voter can then abort the process by disseminating an abort message. The process is a gossiping one when all votes are yes. This implies that *finding the minimum communication cost of a k-ft atomic commitment algorithm is* NP-*hard.*

A k-ft commit algorithm COMMIT($c,\mathscr{P},k$) can be obtained (Fig. 6) by the conjunction of the gossiping algorithm BSM-GOSSIPING($c,\mathscr{P},k$) and of the MULTI-BROADCASTING($c,\mathscr{P},k$) broadcasting algorithm discussed in the previous Sections 3 and 4. Indeed, the definition of the commitment problem implies that a no voter can

COMMIT($c,\mathcal{P},k$) \at participant i

1. Construct a minimum spanning tree T of the cost graph $K_{\mathcal{P}}$.
2. Construct, starting from T, an approximate minimum BSM B and call σ and σ' its source and sink.
3. B_0 is obtained by rooting T in the source σ of B_1 and directing its edges toward σ and
   Each B_i, $1 \le i \le k$ is obtained from B by taking source σ (resp. σ') and sink σ' (σ) if i is odd (even) and directing its arcs from the source to the sink.
   B_{k+1} is obtained by rooting T in the sink of B_k and directing its edges away from the root.
4. Apply the labelling LABEL($B_0, B_1, \ldots, B_{k+1}$).
5. If "no voter" then send the abort message $k+1$ times to each neighbor in T.
6. If "yes voter" then
   Let $t = 0$ and $abort = no$ \\$abort = no$ iff no abort message is arrived
   while ($t \le k+1$ and $abort = no$) do
      as the packets from all incoming arcs in B_t (B_{t-1} if source of B_t) are arrived or the maximum waiting time (label of the outgoing arcs in B_t) is elapsed send along the outgoing arcs in B_t a yes message.
      Set $t = t+1$.
   if $abort = yes$ then \broadcast abort
      Let j be the neighbor from which the first copy of the abort message is received
      each time a copy of the abort message is received from j forward this copy (separately) to each other neighbor in T but those from which an abort message is arrived.

Fig. 6.

immediately start to perform a k-ft broadcasting algorithm to diffuse its abort message. On the other hand a yes voter behaves as a participant in the BSM-GOSSIPING($c, \mathscr{P}, k$) k-ft gossiping algorithm till the end of the algorithm or till it is reached by an abort message; in such a case the participant will continue by executing its role in the k-ft broadcasting algorithm MULTI-BROADCASTING($c, \mathscr{P}, k$) with the no voters as multiple sources.

Results analogous to those of Theorems 5.1, 5.2, and 5.3 can be easily derived for the COMMIT($c, \mathscr{P}, k$).

References

[1] R. Ahlswede, L. Gargano, H.S. Haroutunian, L.H. Khachatrian, Fault-tolerant minimum broadcast networks, Networks 27 (1996) 293–307.

[2] A. Bagchi, S.L. Hakimi, Information dissemination in distributed systems with faulty units, IEEE Trans. Comput. 43 (1994) 698–710.

[3] A. Bagchi, E.F. Schmeichel, S.L. Hakimi, Parallel information dissemination by packets, SIAM J. Comput. 23 (1994) 355–372.

[4] A. Bagchi, E.F. Schmeichel, S.L. Hakimi, Sequential information dissemination by packets, Networks 22 (1992) 317–333.

[5] B. Backer, R. Shostah, Gossips and telephones, Discrete Math. 2 (1972) 191–193.

[6] A. Bar-Noy, J. Bruck, C.T. Ho, S. Kipnis, B. Schieber, Computing global combine operations in the multi-port postal model, Proc. 5th IEEE Symp. on Parallel and Distributed Computing (SPDP 93), Dallas, TX, 1993, pp. 336–343.

[7] K.A. Berman, M. Hawrycz, Telephone problems with failures, SIAM J. Algerbraic Discrete Meth. 7 (1986) 13–17.

[8] J.-C. Bermond, P. Fraigniaud, Broadcasting and gossiping in de Bruijn networks, SIAM J. Comput. 23 (1994) 212–225.

[9] J.-C. Bermond, L. Gargano, A. Rescigno, U. Vaccaro, Optimal gossiping by short messages, SIAM J. Comput., to appear.

[10] D.P. Bertsekas, J.N. Tsitsiklis, Parallel and Distributed Computation: Numerical Methods, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[11] R.T. Bumby, A problem with telephone, SIAM J. Algebraic Discrete Meth. 2 (1981) 13–19.

[12] T. Cormen, C. Leiserson, R. Rivest, Introduction to Algorithms, McGraw-Hill, New York, 1990.

[13] N. Cot, Extensions of the telephone problem, Proc. 7th SE Conf. on Comb. Graph Theory and Computation, Utilitas Mathematica, Winnipeg, Manitoba, Canada, 1976, pp. 239–256.

[14] K. Diks, A. Pelc, Efficient gossiping by packets in networks with random faults, SIAM J. Discrete Math. 9 (1996) 7–18.

[15] D. Dolev, C. Dwork, L. Stockmayer, On the minimal synchronism needed for distributed consensus, J. ACM 34 (1987) 77–97.

[16] J. Edmonds, Edge-disjoint branchings, Combinatorial Algorithms, Algorithmic Press, NY, 1973, 91–96.

[17] K.P. Eswaran, R.E. Tarjan, Augmentation problems, SIAM J. Comput. 5 (1976) 653–665.

[18] S. Even, B. Monien, On the number of rounds necessary to disseminate information, Proc. 1st ACM Symp. on Parallel Algorithms and Architectures, Santa Fe, NM, 1989, pp. 318–327.

[19] P. Fraignaud, E. Lazard, Methods and problems of communication in usual networks, Discrete Appl. Math. 53 (1994) 79–134.

[20] M.R. Garey, D.S. Johnson, Computer and Intractability: A Guide to the Theory of NP-Completeness, Freeman, San Fransisco, 1979.

[21] L. Gargano, Tighter time bounds on fault-tolerant broadcasting and gossiping, Networks 22 (1992) 469–486.

[22] L. Gargano, A. Liestman, J. Peters, D. Richards, Reliable broadcasting, Discrete Appl. Math. 53 (1994) 135–148.

[23] L. Gargano, A. Rescigno, U. Vaccaro, Communication complexity of gossiping by packets, Proc. 5th Scandinavian Workshop on Algorithm Theory (SWAT'96), Reykjavik, Iceland, 1996, pp. 234–245.
[24] L. Gargano, A. Rescigno, U. Vaccaro, Fault-tolerant hypercube broadcasting via information dispersal, Networks 23 (1993) 271–282.
[25] L. Gargano, U. Vaccaro, Minimum time broadcast networks tolerating a logarithmic number of faults, SIAM J. Discrete Math. 5 (1992) 178–198.
[26] V. Hadzilacos, A knowledge theoretic analysis of atomic commitment protocols, Proc. 6th ACM Symp. on Principles of Database Systems (1987) 129–134.
[27] R.W. Haddad, S. Roy, A.A. Schäffer, On gossiping with faulty telephone lines, SIAM J. Algebraic Discrete Meth. 8 (1987) 439–445.
[28] A. Hajnal, E.C. Milner, E. Szemeredi, A cure for telephone diseases, Can. Math. Bull. 15 (1976) 447–450.
[29] F. Harary, A.J. Schwenk, The communication problem on graphs and digraphs, J. Franklin Inst. 297 (1974) 491–495.
[30] S. Hedetniemi, S. Hedetniemi, A. Liestman, A survey of gossiping and broadcasting in communication networks, Networks 18 (1988) 129–134.
[31] J. Hromkovic, C.-D. Jeschke, B. Monien, Note on optimal gossiping in some weak-connected graphs, Theoret. Comput. Sci. 127 (1994) 395–402.
[32] J. Hromkovič, R. Klasing, B. Monien, R. Peine, Dissemination of information in interconnection networks (broadcasting and gossiping), in: F. Hsu, D.-Z. Du (Eds.) Combinatorial Network Theory, Kluwer Academic Publishers (1995) pp. 125–212.
[33] D.J. Kleitman, J.B. Shearer, Further gossip problems, Discrete Math. 30 (1980) 151–156.
[34] W. Knödel, New gossips and telephons, Discrete Math. 30 (1980) 151–156.
[35] D.W. Krumme, Reordered gossip schemes, Discrete Math. 156 (1996) 113–140.
[36] D.W. Krumme, K.N. Venkataraman, G. Cybenko, Gossiping in minimal time, SIAM J. Comput. 21 (1992) 111–139.
[37] R. Labahn, Kernels of minimum size gossip schemes, Discrete Math. 143 (1995) 99–139.
[38] A.L. Liestman, Fault-tolerant broadcast graphs, Networks 15 (1985) 159–171.
[39] L. Lovasz, Combinatorial Problems and Exercises, Elsevier, Amsterdam, 1993.
[40] A. Pelc, Fault tolerant broadcasting and gossiping in communication networks, Networks 28 (1996) 143–156.
[41] D. Peleg, A note on optimal time broadcast in faulty hypercubes, J. Parallel Distrib. Comput. 26 (1995) 132–135.
[42] D. Peleg, A.A. Schäffer, Time bounds on fault-tolerant broadcasting, Networks 19 (1989) 803–822.
[43] R. Ravi, Rapid rumour ramification: approximating the minimum broadcasting time, Proc. 35th Annu. Symp. on Foundations of Computer Science (FOCS '94), 1994, pp. 202–213.
[44] A. Rescigno, Communication complexity of polling, Inform. Process. Lett. 59 (1996) 317–323.
[45] Y. Saad, M.H. Scultz, Data communication in hypercubes, Parallel Computing 11 (1989) 131–150.
[46] Q. Stout, B. Wagar, Intensive hypercube communication, J. Parallel Distrib. Comput. 10 (1990) 167–181.
[47] A.S. Tanenbaum, Computer Networks, Prentice-Hall, Englewood Cliffs, NJ, 1981.
[48] O. Wolfson, A. Segall, The communication complexity of atomic commitment and of gossiping, SIAM J. Comput. 20 (1991) 423–450.